3D - Learning Representations From Audio Using Autoencoders

Bharathi.A [1], Prakash.J[2]
{abi.cse@psgtech.ac.in[1], jpk.cse@psgtech.ac.in[2]}

Assistant Professor, Dept. of Computer Science and Engineering,PSG College of technology,Coimbatore, India[1,2]

Abstract. Deep learning methods permit us to tackle signal processing challenges from a dissimilar perspective, which is currently overlooked in the composition of music in cinema industry. Audio is inherently added time-sensitive than movie. Audios are encoded using other past methods, resulting in data loss or temporal anomalies. This problem is alleviated by using an auto correlogram with a 3-dimensional view, including time, power, and frequency, to improve accuracy. First, acoustic data should be competently encoded into a compressed format using RNN autoencoder by interrelating with the information. As a result of the compressed format, audio waves should be accurately represented. After that, audio waves are rebuilt into an audio structure with little data loss. The accuracy is improved by 10% by using the RNN encoder and decoder.

Keywords: Audio signal, Auto correlogram, RNN encoder, RNN decoder.

1 Introduction

Various customized characteristics calculated from raw audio data are often used in machine learning algorithms for audio processing. As a result, a great deal of effort has gone into emerging high-performing information sets for certain errands. Deep representation learning specific, has lately gotten a lot of press as a very effective substitute to using regular feature sets [1][2]. For many applications, such as speech recognition and music transcription, these approaches outperform feature engineering[2][3][4][17]. On the other hand, deep neural networks struggle with sequential data like audio since they often demand inputs with a fixed dimensionality In this line, machine translation has suggested RNNs (Recurrent Neural Networks) with sequence to sequence learning to understand fixed length illustrations of variable sequences length [5].

Time and accuracy are major problems in today's environment. As a result, any solution that can efficiently cut time and produce more accurate results is recognised and valued. When audios were encoded using different methods in the past, it resulted in data loss or temporal anomalies. As a result, we can provide more accuracy when reconstructing encoded audio files. To learn a representation (encoding), we utilise an auto encoder, and to categorise predictions, we employ LSTM networks. [2-3], [6–8][16].

Previously, auto correlograms were shown in a 2D image with power and time on both axes. This work describes auto correlogram in 3D perspective to increase accuracy, encompassing time, power, and frequency. Raw audio files are compressed using a variety of

ICCAP 2021, December 07-08, Chennai, India

DOI 10.4108/eai.7-12-2021.2314968

audio encoders. However, because of their temporal errors, these encoders cannot recreate the duplicate audio files. This research presents a novel audio compression technique that compresses temporal information while minimising data loss. By interacting with the knowledge/data store, acoustic sequence must be effectively encoded into a crushed format utilising RNN auto encoder. As a result of the compressed format, audio waves should be accurately represented. After that, audio waves are rebuilt into an audio format with little data loss.

This article feeds audio data into an auto correlogram, which displays audio waves in three dimensions: power, time, and frequency. The RNN encoder then gives the output, which conducts consecutive input and output processes in the improving accuracy. The process is continued till the highest level of accuracy is achieved. Furthermore, this encoder can display temporal dynamic behaviour for a time series.

2. Related Work

Nicolas Boulanger-Lewandowski et al [9] investigate the challenge of representing symbolic polyphonic music sequences in a highly generic piano-roll form. Based on recurrent neural network distribution estimators, they provide a probabilistic model for detecting temporal relationships in large dimensional sequences. Their technique outperforms several classic polyphonic music models on a range of actual data sets. They also demonstrate in what way the musical language model shall be cast-off as symbol in increase polyphonic transcription accurateness.

A spectrogram is a graph with three dimensions: time or RPM, frequency, and the frequency amplitude at a specific time. The intensity or colour of each point in the picture represents it. A spectrogram does not contain any information regarding the signal's actual or approximate phase. As a result, reversing the procedure and generating a replica of the original call from a spectrogram is impossible.

Kyunghyun Cho et al[10] proposed Recurrent Neural Network Encoder-Decoder, a neural network architecture comprising two recurrent neural networks. The first RNN converts a set of symbols into a fixed-length vector visualisation, while the second RNN decodes the translation into a new set of symbols. This model's decoder and the encoder is trained together in optimised conditional probability order provided by the sequence source. Using the conditional probabilities of phrase pairings generated by the RNN Encoder-Decoder as an extra feature in the current log-linear model, the performance of the statistical machine translation system is shown to increase empirically.

Shahin Amiriparian et al[3]. For unsupervised representation learning, propose a recurrent sequence to sequence autoencoder. They begin by extracting mel spectrograms from raw audio data. Second, they use these spectrograms in training recurrent seq to seq autoencoder, a time-dependent frequency vector. Third, the learned representations of spectrograms are then extracted as feature vectors for the relevant audio occurrences in fully linked layer among decoder and encoder units. Finally, they use these feature vectors to train a multilayer perceptron neural network to predict class labels.

Zhuotun Zhu et al[11][12]intended to utilize an autoencoder to acquire a 3-dimensional object representation based on projected depth pictures. A 3-dimensional form is transmitted into a variety of depth pictures, which the learned autoencoder can elegantly recreate. The autoencoder-based 3-dimensional form model is a deep learning representation; that is a general representation instead of representations based on local descriptors, such as

SIFT. This deep learning model and the local descriptors-based representation are complimentary.

G. E. Hinton et al[13][3]present a method for effectively initializing the weights in deep auto - encoder networks such that they may learn low-dimensional codes that are far more successful than PCA in reducing data dimensionality. High dimensional information may be changed to low dimensional codes through training a multilayer network with small core layer to replicate high dimensional input vectors. Gradient descent may be cast-off to fine tune weights these "autoencoder" networks, but lone if initial weights remain close to decent fit.

3. Proposed Methodology

The schematic illustration of approach that is proposed are revealed in Figure 1. First, the input audio signal is given as input to the auto correlogram generator through the audio inputter. Next, the auto correlogram with three dimensional, namely time, frequency and frequency amplitude, will be generated. Next, the auto correlogram will be sent through the RNN encoder, producing a compressed audio file. When the compressed audio file is decoded, an original audio file is created and a denoised audio signal.

3.1. Signal Processing:

The fundamental frequency is computed using the Fast Fourier Transform (fft), Fundamental frequency is chosen when fft are at its maximum. When detrend (normalisation) is conducted before fft, the most might occur at zero frequency.

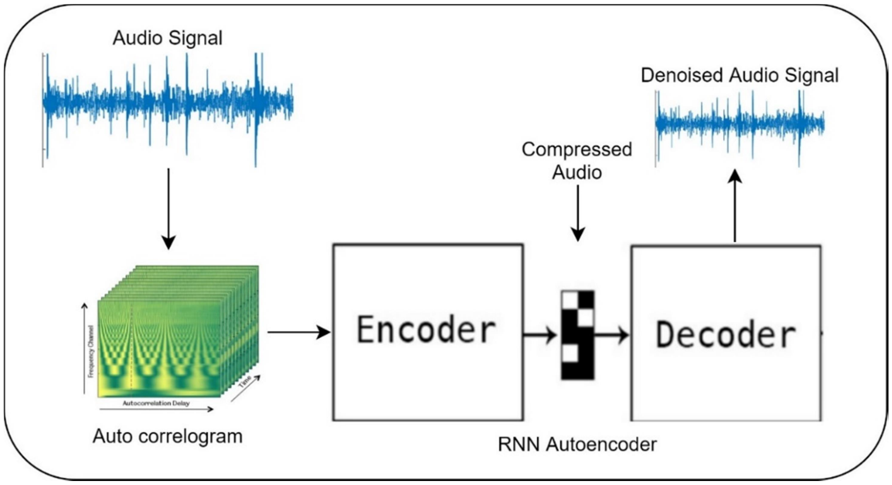


Fig. 1 Schematic Representation of The Proposed System

3.1.1. FFT Algorithm:

The the frequency and the time domains of composite notation, there is a single signal made of N complex points. Two numbers make up the imaginary parts and the real parts in each of its difficult points. For example, For about difficult sample X, we are talking about the grouping of ImX and ReX. In additional words, every complex variable contains 2 integers. If two complex variables is integrated, Product's two components should are made up of four independent components. Single terms such as value, sample, point, and signal refer to the blend of imaginary and real portions.

3.1.1.1.Steps of FFT Algorithm:

i. The Fast Fourier Transform breaks down an N point time domain signal to a N single-point signals.

ii. In second stage, the N frequency spectra associated with N time domain signals are measured. After then, N spectra is combined to form sole frequency spectrum. For FFT time-domain decomposition, a bit reversal sorting technique is typically utilised.

iii. The FFT approach then proceeds to frequency spectra of one point time domain data. Because each point signals is now frequency spectrum rather than time-domain signal, this is case.

iv. FFT's final step to integrate the N frequency spectra in same order that they were decomposed in the time domain.

3.1.2. Fundamental Frequency

It It is usually advantageous to express a specified sinusoidal or complex exponential signal in one of the following formats to determine its period, frequency, or angular frequency:

$$\sin(\omega t) = \sin(2\pi f t) = \sin(2\pi t/T)$$

The fundamental frequency are the GCD of wholly frequency components present in the signal, and essential period is LCM of wholly distinct periods of components.

3.2. Autocorrelation:

As a function of delay, autocorrelation correlates the signal with a delayed replica of itself. Nonchalantly, it is the comparison of comments as function of their time lag. Figure 2 shows auto correlogram that was developed. The autocorrelation analysis is numerical approach in discovering reiterating patterns, for example the detection of signal's missing fundamental frequency, which is indicated by its harmonic frequencies, or the appearance of periodic signals hidden by noise. It is often castoff in signal analysis in look at functions or data sequences, such as time domain signals.

Auto correlation of real or sophisticated random process is a Pearson correlation among the process's values at various times as a function of the two times or the time lag. Let $\{X_t\}$ denote the random process, while t denotes the time interval . Therefore, the value (or realisation) produced by a particular operation execution for time t is $\{X_t\}$. Assume that at time t, is a mean and variance of the process σ_t^2. The auto correlation function among times t_1 and t_2 is then defined as.

$$R_{XX}(t_1, t_2) = E[X_{t_1} X_{t_2}{}']$$

where $E$is the expected value operator. Subtracting the mean before multiplication yields the auto-covariance function between times $t_1$and t_2:

$$K_{XX}(t_1, t_2) = E\big[(X_{t_1} - \mu_{t_1})(X_{t_2} - \mu_{t_2})'\big] = E\big[X_{t_1} X_{t_2}{}'\big] - \mu_{t_1} \mu_{t_2}{}'$$

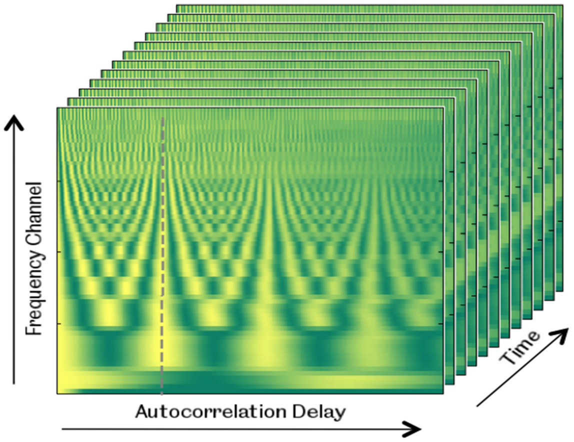


Fig 2 Auto Correlogram

3.3. RNN Autoencoders:

The framework are made up of two modules: one to read input sequence (auto correlogram) and to encode fixed-length vector (Compressed audio), while the other to decode fixed length vector and output anticipated sequence (i.e., Denoised Audio signal). Encoder Decoder LSTM developed particularly for seq2seq issues is moniker given to the framework for usage of the modules. The input and output layers of this LSTM are fully coupled, however because the encoder produces non-linearity, the activation functions used by the input and output layers may differ. At the same time, decoder will employ the linear projection to match the sequences of output and input.

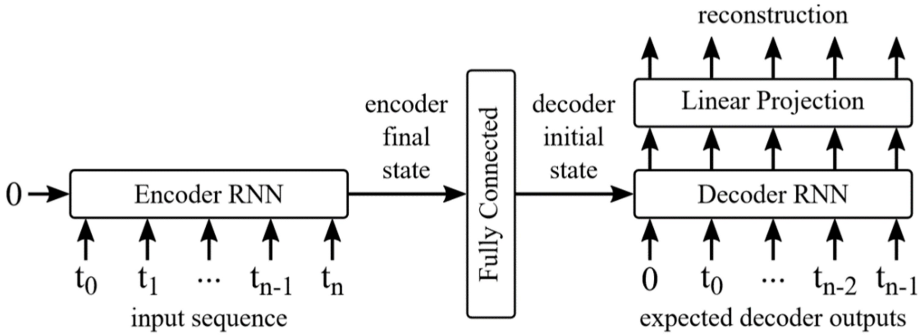


Fig 3. Rnn Encoder And Decoder

4 Experimental Analysis

The python tool kit Deep Spectrum is used to extract features from audio. The RNN encoder-decoder is implemented using keras and tensorflow.

4.1. Dataset:

ESC-50 (Environmental Sound Classification)[14] The dataset contains 2000 tagged environmental records that are uniformly distributed over 50 classes (i.e. 40 clips each class). For the sake of convenience, animal sounds, water sounds, natural soundscapes, inside/local resonances, human sounds, and outside/urban resonances have been split to five loosely defined main groupings (i.e., ten classes per category). The extraction algorithm attempted to keep sound occurrences in front with slight background clatter wherever possible. Ground footages, on the other hand, are distant after sterile. As a result, some recordings may still have audible background overlay. The collection exposures viewers to a variety of sound foundations, some of which are fairly common (dog barking, cat meowing, laughing), others of which are rather distinctive (brushing teeth, glass shattering), and finally, there's a spot where some distinctions and more subtle differences may be found (airplane noise and helicopter) [15]

4.2. Data Model Structure:

Data sets store instances, which can correspond to either an entire audio file or a chunk of an audio file. For each instance, the following attributes represented in Table 1 are stored.

Attribute (Variable Name)	**Value Required**	**Dimensionality**	**Description**
Filename (FILENAME)	Yes	-	The name of the audio file from which the instance was extracted
Chunk Number (CHUNK_NR)	Yes	-	The index of the chunk which the instance represents. The filename and the chunk number attributes together uniquely identify instances.
Nominal Label (LABEL_NOMINAL)	No	-	Nominal label of the instance. If specified, the numeric label must be specified as well.
Numeric Label (LABEL_NUMERIC)	No	-	Numeric label of the instance. If specified, the nominal label must be specified as well.
Cross validation folds (CV_FOLDS)	Yes	Number of Folds	Specifies cross validation information. For each cross validation fold, this attribute stores whether the instance belongs to the training split (0), or the validation split (1). We have chosen to

			represent cross validation information in this way, since we have encountered data sets with overlapping cross validation folds, which can not be represented by simply storing the fold number for each instance. Please note that, while this attribute is required to have a value, this value is allowed to have dimension zero, corresponding to no cross validation information.
Partition (PARTITION)	No	-	The partition to which the instance belongs (0: training, 1: development, 2: test)
Features (FEATURES)	Yes	Arbitrary	The feature matrix of the instance

TABLE 1: ESC-50 data model structure.

4.3 Results:

The audio signal of a ogg file “1-4211-A.ogg” of class “Fire Cackling” remains considered toward the audio wave signal besides the auto correlogram. The same is shown in Figure 4 and figure 5. The acoustic sign of a ogg file “1-4211-A.ogg” of class “Fire Cackling” is designed in Figure 4 to perceive the amplitude and time(sec).

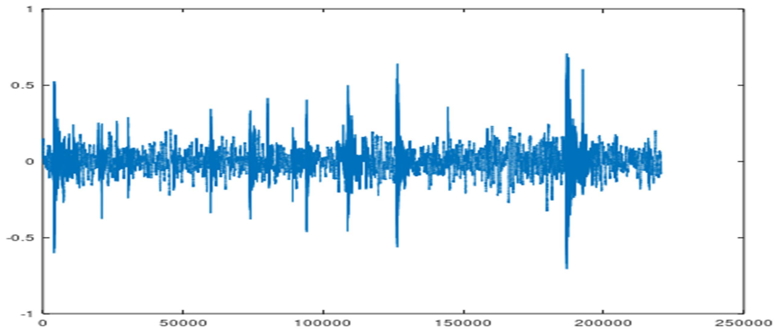


Fig 4. “Fire Cackling” Audio Wave Signal

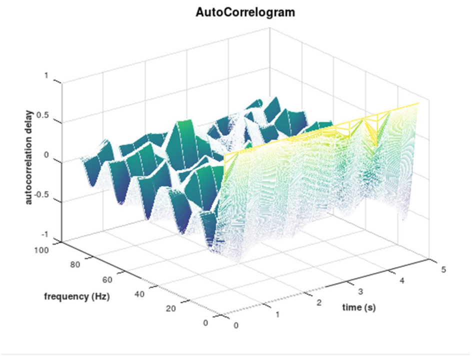
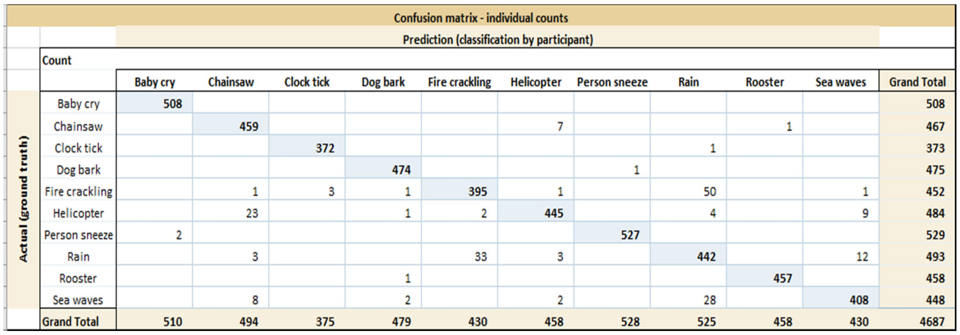


Fig 5: Autocorelogram of “1-4211-A.ogg” of class “Fire Cackling”

4.4. Analysis:

Figure 6 depicts the prediction confusion matrix. The projected class is represented by the column, whereas the actual class is shown by the row. The diagonal represents the correct prediction of each class. The precision of the predicted class is shown in figure 7.

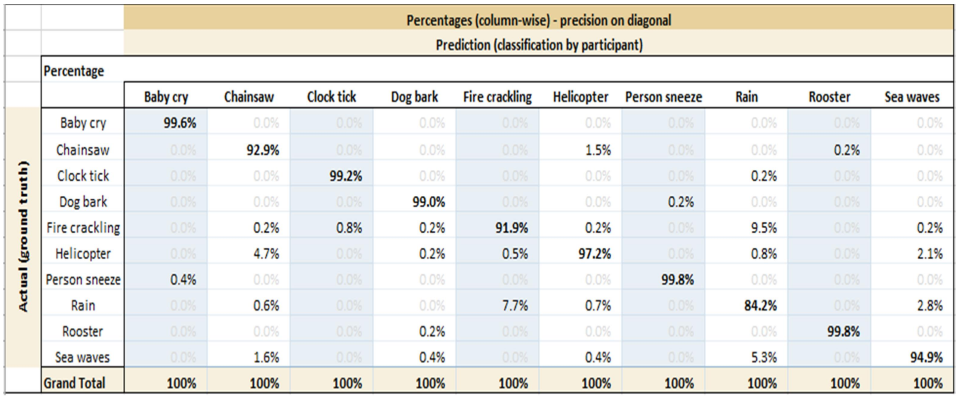
Figure 7 shows that when participants chose the chainsaw class, 92.9 percent of the predictions were for chainsaw recordings. The existing method has an accuracy of 67% on the K-NN method,72% on the Random forest method, and 70% on the SVN method on average. Thus, this technique provided nearly 78% accuracy on average. Compared with previous approaches, the accuracy rate has been improved by more than 10%, as shown in Figure 8.



Confusion matrix - individual counts

		Prediction (classification by participant)										
	Count											
		Baby cry	Chainsaw	Clock tick	Dog bark	Fire crackling	Helicopter	Person sneeze	Rain	Rooster	Sea waves	Grand Total
Actual (ground truth)	Baby cry	508										508
	Chainsaw		459				7			1		467
	Clock tick			372					1			373
	Dog bark				474			1				475
	Fire crackling		1	3	1	395	1		50		1	452
	Helicopter		23		1	2	445		4		9	484
	Person sneeze	2						527				529
	Rain		3			33	3		442		12	493
	Rooster				1					457		458
	Sea waves		8		2		2		28		408	448
	Grand Total	510	494	375	479	430	458	528	525	458	430	4687

Fig 6: Confusion Matrix of Prediction



Percentages (column-wise) - precision on diagonal

Prediction (classification by participant)

	Percentage	Baby cry	Chainsaw	Clock tick	Dog bark	Fire crackling	Helicopter	Person sneeze	Rain	Rooster	Sea waves
Actual (ground truth)	Baby cry	99.6%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
	Chainsaw	0.0%	92.9%	0.0%	0.0%	0.0%	1.5%	0.0%	0.0%	0.2%	0.0%
	Clock tick	0.0%	0.0%	99.2%	0.0%	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%
	Dog bark	0.0%	0.0%	0.0%	99.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%
	Fire crackling	0.0%	0.2%	0.8%	0.2%	91.9%	0.2%	0.0%	9.5%	0.0%	0.2%
	Helicopter	0.0%	4.7%	0.0%	0.2%	0.5%	97.2%	0.0%	0.8%	0.0%	2.1%
	Person sneeze	0.4%	0.0%	0.0%	0.0%	0.0%	0.0%	99.8%	0.0%	0.0%	0.0%
	Rain	0.0%	0.6%	0.0%	0.0%	7.7%	0.7%	0.0%	84.2%	0.0%	2.8%
	Rooster	0.0%	0.0%	0.0%	0.2%	0.0%	0.0%	0.0%	0.0%	99.8%	0.0%
	Sea waves	0.0%	1.6%	0.0%	0.4%	0.0%	0.4%	0.0%	5.3%	0.0%	94.9%
	Grand Total	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

Fig 7: Precision of Predicted Class

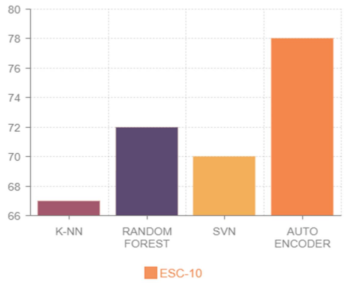


Fig 8: Accuracy Comparison of Various Methods

5. Conclusion

This study delivers improved accuracy when reconstructing encoded audio files. In addition, the novel auto correlogram approach, which offers a three-dimensional picture of the audio wave, was used to compress an audio file. As a consequence, the resulting audio wave will be more accurate than the previous auto correlogram technique's two-dimensional picture. The accuracy rate of the previous system is 70 to 75 percent, but when utilising this method, the accuracy rate is 75 to 80 percent, a nearly 10% improvement. The accuracy rate can be enhanced more in the future by learning representations from a large dataset.

References

[1] Y. Bengio, A. C. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.

[2] Y. Bengio, N. Boulanger-Lewandowski, and R. Pascanu, “Advances in optimizing recurrent networks,” in 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, 2013, pp. 8624–8628. doi: 10.1109/ICASSP.2013.6639349.

[3] S. Amiriparian, P. Winokurow, V. Karas, S. Ottl, M. Gerczuk, and B. Schuller, “Unsupervised Representation Learning with Attention and Sequence to Sequence Autoencoders to Predict Sleepiness From Speech,” in Proceedings of the 1st International on Multimodal Sentiment Analysis in Real-Life Media Challenge and Workshop, 2020, pp. 11–17. doi: 10.1145/3423327.3423670.

[4] S. Amiriparian, J. Pohjalainen, E. Marchi, S. Pugachevskiy, and B. W. Schuller, “Is Deception Emotional? An Emotion-Driven Predictive Approach,” in Interspeech 2016, 17th Annual Conference of the International Speech Communication Association, San Francisco, CA, USA, September 8-12, 2016, 2016, pp. 2011–2015. doi: 10.21437/Interspeech.2016-565.

[5] I. Sutskever, O. Vinyals, and Q. v Le, “Sequence to Sequence Learning with Neural Networks,” in Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, 2014, pp. 3104–3112.

[6] H. Lee, Y. Largman, P. Pham, and A. Y. Ng, “Unsupervised Feature Learning for Audio Classification Using Convolutional Deep Belief Networks,” in Proceedings of the 22nd International Conference on Neural Information Processing Systems, 2009, pp. 1096–1104.

[7] Y.-A. Chung, C.-C. Wu, C.-H. Shen, and H. Lee, “Unsupervised Learning of Audio Segment Representations using Sequence-to-sequence Recurrent Neural Networks,” 2016.

[8] Y. Aytar, C. Vondrick, and A. Torralba, “SoundNet: Learning Sound Representations from Unlabeled Video,” in Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 892–900.

[9] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent, “Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription,” in Proceedings of the 29th International Coference on International Conference on Machine Learning, 2012, pp. 1881–1888.

[10] K. Cho et al., “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1724–1734. doi: 10.3115/v1/d14-1179.

[11] Z. Zhu, X. Wang, S. Bai, C. Yao, and X. Bai, “Deep learning representation using autoencoder for 3D shape retrieval,” Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC), pp. 279–284, 2014.

[12] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, “Unsupervised Learning of Sparse Features for Scalable Audio Classification,” in Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011, Miami, Florida, USA, October 24-28, 2011, 2011, pp. 681–686. [Online]. Available: http://ismir2011.ismir.net/papers/PS6-5.pdf

[13] G. E. Hinton and R. R. Salakhutdinov, “Reducing the dimensionality of data with neural networks,” Science, vol. 313, no. 5786, pp. 504–507, Jul. 2006, doi: 10.1126/science.1127647.

[14] K. J. Piczak, “ESC: Dataset for Environmental Sound Classification,” in Proceedings of the 23rd ACM International Conference on Multimedia, 2015, pp. 1015–1018. doi: 10.1145/2733373.2806390.

[15] K. J. Piczak, “Environmental sound classification with convolutional neural networks,” in 2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP), 2015, pp. 1–6. doi: 10.1109/MLSP.2015.7324337.

[16] Aditya Kumar Yadav, & Prakash, J. (2021). Image Captioning Using R-CNN & LSTM Deep Learning Model. International Journal of Innovative Science and Research Technology, 6(5), 911-914.

[17] Predicting flight delay using ANN with multi-core map reduce framework. (2016). Communication and Power Engineering, 280-287. https://doi.org/10.1515/9783110469608-028